Cycle killer... Qu'est-ce que c'est? On the Comparative Approximability of Hybridization Number and Directed Feedback Vertex Set

Kelk, S.; van Iersel, L.J.J.; Lekic, N.; Linz, S.; Scornavacca, C.; Stougie, L.

published in
SIAM Journal on Discrete Mathematics
2012

DOI (link to publisher)
10.1137/120864350

document version
Publisher's PDF, also known as Version of record

SIAM J. DISCRETE MATH.
Vol. 26, No. 4, pp. 1635–1656

CYCLE KILLER ... QU'EST-CE QUE C'EST? ON THE COMPARATIVE APPROXIMABILITY OF HYBRIDIZATION NUMBER AND DIRECTED FEEDBACK VERTEX SET*

STEVEN KELK†, LEO VAN IERSEL‡, NELA LEKIĆ†, SIMONE LINZ§, CELINE SCORNAVACCA¶, AND LEEN STOUGIE‖

Abstract. We show that the problem of computing the hybridization number of two rooted binary phylogenetic trees on the same set of taxa X has a constant factor polynomial-time approximation if and only if the problem of computing a minimum-size feedback vertex set in a directed graph (DFVS) has a constant factor polynomial-time approximation. The latter problem, which asks for a minimum number of vertices to be removed from a directed graph to transform it into a directed acyclic graph, is one of the problems in Karp's seminal 1972 list of 21 NP-complete problems. Despite considerable attention from the combinatorial optimization community, it remains to this day unknown whether a constant factor polynomial-time approximation exists for DFVS. Our result thus places the (in)approximability of hybridization number in a much broader complexity context, and as a consequence we obtain that it inherits inapproximability results from the problem VERTEX COVER. On the positive side, we use results from the DFVS literature to give an $O(\log r \log\log r)$ approximation for the hybridization number where r is the correct value.

Key words. hybridization number, phylogenetic networks, directed feedback vertex set, approximation

AMS subject classifications. 68W25, 05C20, 90C27, 92B10

DOI. 10.1137/120864350

1. Introduction.

The traditional model for representing the evolution of a set of species X (or, more generally, a set of *taxa*) is the *rooted phylogenetic tree* [17, 18, 36]. Essentially, this is a rooted tree where the leaves are bijectively labeled by X and the edges are directed away from the unique root. A *binary* rooted phylogenetic tree carries the additional restriction that the root has indegree zero and outdegree two, leaves have indegree one and outdegree zero, and all other (internal) vertices have indegree one and outdegree two. Rooted binary phylogenetic trees will have a central role in this article.

In recent years there has been a growing interest in extending the phylogenetic tree model to also incorporate nontreelike evolutionary phenomena such as hybridizations,

*Received by the editors February 1, 2012; accepted for publication (in revised form) August 10, 2012; published electronically November 20, 2012. Part of this work was conducted at the Isaac Newton Institute for Mathematical Sciences in Cambridge, England, in June 2011.
http://www.siam.org/journals/sidma/26-4/86435.html
†Department of Knowledge Engineering (DKE), Maastricht University, 6200 MD Maastricht, The Netherlands (steven.kelk@maastrichtuniversity.nl, nela.lekic@maastrichtuniversity.nl). The work of the third author was supported by a Vrije Competitie grant of The Netherlands Organisation for Scientific Research (NWO).
‡Centrum Wiskunde and Informatica (CWI), 1090 GB Amsterdam, The Netherlands (l.j.j.v.iersel@gmail.com). The work of this author was supported by a Veni grant of NWO.
§Center for Bioinformatics (ZBIT), Tübingen University, 72076 Tübingen, Germany (linz@informatik.uni-tuebingen.de). This author was partially supported by a Marie Curie International Outgoing Fellowship within the 7th European Community Framework Programme.
¶Institut des Sciences de l'Evolution (ISEM, UMR 5554 CNRS), Université Montpellier II, 34095 Montpellier Cedex 5, France (celine.scornavacca@univ-montp2.fr).
‖CWI and Operations Research, Department of Economics and Business Administration, Vrije Universiteit, 1081 HV Amsterdam, The Netherlands (l.stougie@vu.nl). The work of this author was supported by the CLS/NWO MEMESA project and the Tinbergen Institute.

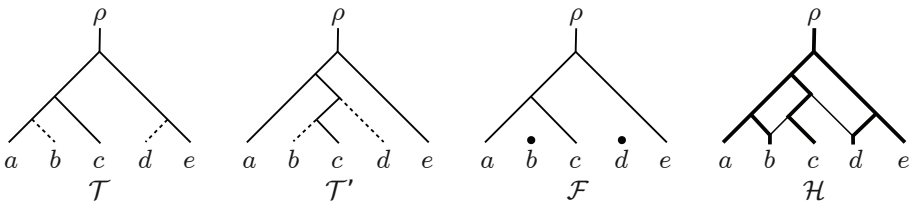

FIG. 1.1. *Two phylogenetic trees, $\mathcal{T}$ and $\mathcal{T}'$, an acyclic agreement forest $\mathcal{F}$ for $\mathcal{T}$ and $\mathcal{T}'$ and a hybridization network $\mathcal{H}$ that displays $\mathcal{T}$ and $\mathcal{T}'$ and has hybridization number 2. All edges are directed downwards. Forest $\mathcal{F}$ can be obtained from either of $\mathcal{T}$ and $\mathcal{T}'$ by deleting the dashed edges. Bold edges are used in $\mathcal{H}$ to illustrate that this network displays $\mathcal{T}$. The function of the vertex labeled ρ is explained in the preliminaries.*

recombinations, and horizontal gene transfers. This has stimulated research into *rooted phylogenetic networks*, which generalize rooted phylogenetic trees by also permitting vertices with indegree two or higher, called *reticulation* vertices (or *hybridization* vertices). Reticulation vertices with indegree greater than two represent multiple reticulate evolutionary events. The number of *reticulations* specified by a reticulation vertex is equal to its indegree minus one. For detailed background information on phylogenetic networks we refer the reader to [22, 23, 24, 38, 30, 35]. In a rooted *binary* phylogenetic network the reticulation vertices all have indegree two and outdegree one (and all other vertices obey the usual restrictions of a rooted binary phylogenetic tree).

Informally, we say that a phylogenetic network $\mathcal{N}$ on X *displays* a phylogenetic tree $\mathcal{T}$ on X if it is possible to delete all but one incoming edge of each reticulation vertex of $\mathcal{N}$ such that, after subsequently suppressing vertices which have indegree and outdegree both equal to one, the tree $\mathcal{T}$ is obtained (see Figure 1.1). Following the publication of several seminal articles in 2004–2005 (e.g., [2, 3]), there has been considerable research interest in the following biologically inspired question. Given two rooted, binary phylogenetic trees $\mathcal{T}$ and $\mathcal{T}'$ on the same set of taxa X, what is the minimum number of reticulations required by a phylogenetic network $\mathcal{N}$ on X which displays both $\mathcal{T}$ and $\mathcal{T}'$? This value is often called the *hybridization number* in the literature, and when addressing this specific problem the term *hybridization network* is often used instead of the more general term phylogenetic network. For the purpose of consistency we will henceforth use the term hybridization network in this article.

MINIMUMHYBRIDIZATION, the problem of computing the hybridization number seen as a minimization problem, has been shown to be both NP-hard and APX-hard [8], from which several related phylogenetic network construction techniques also inherit hardness [28, 38]. APX-hardness means that there exists a constant $c > 1$ such that the existence of a polynomial-time approximation algorithm that achieves an approximation ratio strictly smaller than c would imply P = NP. As is often the case with APX-hardness results, the value c given in [8] is very small, $\frac{2113}{2112}$. It is not known whether MINIMUMHYBRIDIZATION is actually in APX, the class of problems for which polynomial-time approximation algorithms exist that achieve a constant approximation ratio. In fact, there are to date no nontrivial polynomial-time approximation algorithms, constant factor or otherwise, for MINIMUMHYBRIDIZATION. This omission stands in stark contrast to other positive results, which we now discuss briefly.

On the fixed parameter tractability front—we refer the reader to [14, 16, 19, 31] for an introduction—a variety of increasingly sophisticated algorithms have been developed. These show that for many practical instances of MINIMUMHYBRIDIZATION

the problem can be efficiently solved [5, 7, 10, 11, 34, 40, 41]. Second, the problem of computing the *rooted subtree prune and regraft* (rSPR) distance, which bears at least a superficial similarity to the computation of hybridization number, permits a polynomial-time 3-approximation algorithm [6, 33, 40] and efficient fixed parameter tractable (FPT) algorithms [6, 39, 40]. Why, then, is it so difficult to give formal performance guarantees for approximating MinimumHybridization?

A clue lies in the nature of the abstraction that (with very few exceptions) is used to compute hybridization number, the *maximum acyclic agreement forest* (MAAF), introduced in [2] (see Figure 1.1). Roughly speaking, computing the hybridization number of two trees $\mathcal{T}$ and $\mathcal{T}'$ is essentially identical to the problem of cutting $\mathcal{T}$ and $\mathcal{T}'$ into as few vertex-disjoint subtrees as possible such that (i) the subtrees of $\mathcal{T}$ are isomorphic to the subtrees of $\mathcal{T}'$ and—critically—(ii) a specific "reachability" relation on these subtrees is acyclic. Condition (ii) is the core of the issue, because without this condition the problem would be no different to the problem of computing the rSPR distance, which, as previously mentioned, seems to be comparatively tractable. (Note that the hybridization number of two trees can in general be much larger than their rSPR distance [21].) The various FPT algorithms for computing hybridization number deal with the unwanted cycles in the reachability relation in a variety of ways, but all resort to some kind of brute force analysis to optimally avoid (e.g., [34]) or break (e.g., [10, 40]) them.

In this article we demonstrate why it is so difficult to deal with the cycles. It turns out that MinimumHybridization is, in an approximability sense, a close relative of the problem Feedback Vertex Set on directed graphs (DFVS). In this problem we wish to remove a minimum number of vertices from a directed graph to transform it into a directed acyclic graph. DFVS belongs to Karp's famous 1972 list of 21 NP-complete problems [26] and is also known to be APX-hard [25]. However, despite almost forty years of attention it is still unknown whether DFVS permits a constant approximation ratio, i.e., whether it is in APX. (The undirected variant of FVS, in contrast, appears to be significantly more tractable. It is 2-approximable even in the weighted case [1].)

By coupling the approximability of MinimumHybridization to DFVS we show that MinimumHybridization is just as hard as a problem that has so far eluded the entire combinatorial optimization community. Specifically, we show that for every constant $c > 1$ and every $\epsilon > 0$ the existence of a polynomial-time c-approximation for MinimumHybridization would imply a polynomial-time $(c + \epsilon)$-approximation for DFVS. In the other direction we show that, for every $c > 1$, the existence of a polynomial-time c-approximation for DFVS would imply a polynomial-time $6c$-approximation for MinimumHybridization. In other words, DFVS is in APX if and only if MinimumHybridization is in APX. Hence a constant factor approximation algorithm for either problem would be a major breakthrough in theoretical computer science.

There are several interesting spin-off consequences of this result, both negative and positive. On the negative side, it is known that there is a very simple parsimonious reduction from the classical problem Vertex Cover to DFVS [26]. Consequently, a c-approximation for DFVS entails a c-approximation for Vertex Cover for every $c \geq 1$. For $c < 10\sqrt{5} - 21 \approx 1.3606$ there cannot exist a polynomial-time c-approximation of Vertex Cover, assuming P $\neq$ NP [12, 13]. Also, if the Unique Games Conjecture is true, then for $c < 2$ there cannot exist a polynomial-time c-approximation of Vertex Cover [29]. (Whether Vertex Cover permits a constant factor approximation ratio strictly smaller than 2 is a long-standing open

problem.) The main result in this article hence not only shows that MINIMUMHYBRIDIZATION is in APX if and only if DFVS is in APX, but also that MINIMUMHYBRIDIZATION cannot be approximated within a factor of 1.3606, unless P = NP (and not within a factor smaller than 2 if the Unique Games Conjecture is true). This improves significantly on the current APX-hardness threshold of $\frac{2113}{2112}$.

On the positive side, we observe that already-existing approximation algorithms for DFVS can be utilized to give asymptotically comparable approximation ratios for MINIMUMHYBRIDIZATION. To date the best polynomial-time approximation algorithms for DFVS achieve an approximation ratio of $O(\min\{\log n \log\log n, \log \tau^* \log\log \tau^*\})$, where n is the number of vertices in the graph and τ^* is the optimal fractional solution of the problem (taking the weights of the vertices into account) [15, 37]. We show that this algorithm can be used to give an $O(\log r \log\log r)$-approximation algorithm for MINIMUMHYBRIDIZATION, where r is the hybridization number of the two input trees. To the best of our knowledge, this is the first nontrivial polynomial-time approximation algorithm for MINIMUMHYBRIDIZATION.

The main result also has interesting consequences for the fixed parameter tractability of MINIMUMHYBRIDIZATION. The inflation factor of 6 in the reduction from DFVS to MINIMUMHYBRIDIZATION is very closely linked to a reduction described by Bordewich and Semple [7]. They showed that the input trees can be reduced to produce a weighted instance containing at most $14r$ taxa. (The fact that the reduced instance is weighted means that it cannot be automatically used to obtain a constant-factor approximation algorithm.) In this article we sharpen their analysis to show that the reduction they describe actually produces a weighted instance with at most $9r$ taxa. Without this sharpening, the inflation factor we obtain would have been higher than 6. From this analysis it becomes clear that the kernel size has an important role to play in analyzing the approximability of MINIMUMHYBRIDIZATION.

This raises some interesting general questions about the linkages between MINIMUMHYBRIDIZATION and DFVS. For example, the reduction by Bordewich and Semple, which gives a linear kernel for a weighted variant of MINIMUMHYBRIDIZATION, can be modified slightly (as is done, for example, in [4] for unrooted SPR distance) to obtain a quadratic kernel for MINIMUMHYBRIDIZATION (without weights). This contrasts sharply with DFVS. It is known that DFVS is FPT [9], but it is *not* known whether DFVS permits a polynomial-size kernel. Might MINIMUMHYBRIDIZATION give us new insights into the structure of DFVS (and vice versa)? More generally, within which complexity frameworks is one of the two problems strictly harder than the other?

The structure of this article is as follows. In the next section, we define the considered problems formally and describe the reductions that were used to show that MINIMUMHYBRIDIZATION is FPT. In section 3, we show an improved bound on the sizes of reduced instances. Subsequently, we use these results to show an approximation-preserving reduction from MINIMUMHYBRIDIZATION to DFVS in section 4 and an approximation-preserving reduction from DFVS to MINIMUMHYBRIDIZATION in section 5.

2. Preliminaries.

Phylogenetic trees. Throughout the paper, let X be a finite set of *taxa* (taxonomic units). A *rooted binary phylogenetic X-tree* $\mathcal{T}$ is a rooted tree whose root has degree two, whose interior vertices have degree three, and whose leaves are bijectively labeled by the elements of X. The edges of the tree can be seen as being directed away from the root. The set of leaves of $\mathcal{T}$ is denoted as $\mathcal{L}(\mathcal{T})$. We identify each leaf

with its label. We sometimes call a rooted binary phylogenetic X-tree a *tree* for short. To synchronize with the agreement forest literature, and without loss of generality, it will be helpful to assume that a tree always has an extra vertex, labeled $\rho \notin X$, which is connected to the original root of the tree (see, e.g., Figure 1.1). We then define $\mathcal{L}(\mathcal{T}) = X \cup \{\rho\}$.

In the course of this paper, different types of subtrees play an important role. Let $\mathcal{T}$ be a rooted phylogenetic X-tree and X' a subset of $\mathcal{L}(\mathcal{T})$. The minimal rooted subtree of $\mathcal{T}$ that connects all leaves in X' is denoted by $\mathcal{T}(X')$. Furthermore, the tree obtained from $\mathcal{T}(X')$ by suppressing all vertices with indegree and outdegree both equal to 1 is the *restriction of* $\mathcal{T}$ *to* X' and is denoted by $\mathcal{T}|X'$. When constructing subtrees $\mathcal{T}(X')$ and $\mathcal{T}|X'$ we include ρ if and only if $\rho \in X'$. Lastly, a subtree of $\mathcal{T}$ is *pendant* if it can be detached from $\mathcal{T}$ by deleting a single edge.

Hybridization networks. A *hybridization network* $\mathcal{H}$ on a set X is a rooted acyclic directed graph, which has a single root of outdegree at least 2, has no vertices with indegree and outdegree both 1, and in which the vertices of outdegree 0 are bijectively labeled by the elements of X. A hybridization network is *binary* if all vertices have indegree and outdegree at most 2 and every vertex with indegree 2 has outdegree 1.

As with trees we again assume, without loss of generality, that a hybridization network has an extra vertex labeled ρ which is connected to the original root of the network.

For each vertex v of $\mathcal{H}$, we denote by $d^-(v)$ and $d^+(v)$ its indegree and outdegree, respectively. If (u, v) is an arc of $\mathcal{H}$, we say that u is a *parent* of v and that v is a *child* of u. Furthermore, if there is a directed path from a vertex u to a vertex v, we say that u is an *ancestor* of v and that v is a *descendant* of u.

A vertex of indegree greater than 1 represents an evolutionary event in which lineages combined, such as a hybridization, recombination, or horizontal gene transfer event. We call these vertices *hybridization vertices*. To quantify the number of hybridization events, the *hybridization number* of a hybridization network $\mathcal{H}$ is given by

$$h(\mathcal{H}) = \sum_{v \neq \rho} (d^-(v) - 1).$$

Observe that $h(\mathcal{H}) = 0$ if and only if $\mathcal{H}$ is a tree.

Let $\mathcal{H}$ be a hybridization network on X and $\mathcal{T}$ a rooted binary phylogenetic X'-tree with $X' \subseteq X$. We say that $\mathcal{T}$ is *displayed* by $\mathcal{H}$ if $\mathcal{T}$ can be obtained from $\mathcal{H}$ by deleting vertices and edges and suppressing vertices with $d^+(v) = d^-(v) = 1$ (or, in other words, if a subdivision of $\mathcal{T}$ is a subgraph of $\mathcal{H}$). Intuitively, if $\mathcal{H}$ displays $\mathcal{T}$, then all of the ancestral relationships visualized by $\mathcal{T}$ are visualized by $\mathcal{H}$.

The problem MinimumHybridization is to compute the *hybridization number* of two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$, which is defined as

$$h(\mathcal{T}, \mathcal{T}') = \min\{h(\mathcal{H}) : \mathcal{H} \text{ is a hybridization network that displays } \mathcal{T} \text{ and } \mathcal{T}'\},$$

i.e., the minimum number of hybridization events necessary to display two rooted binary phylogenetic trees.

This problem can be formulated as an optimization problem in the obvious way.
Problem: MinimumHybridization.
Instance: Two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$.
Solution: A hybridization network $\mathcal{H}$ that displays $\mathcal{T}$ and $\mathcal{T}'$.
Objective: Minimize $h(\mathcal{H})$.

If $\mathcal{H}$ is a hybridization network that displays $\mathcal{T}$ and $\mathcal{T}'$, then there also exists a binary hybridization network $\mathcal{H}'$ that displays $\mathcal{T}$ and $\mathcal{T}'$ such that $h(\mathcal{H}) = h(\mathcal{H}')$ [38, Lemma 3]. Hence, we restrict our analysis to binary hybridization networks and will not emphasize again that we only deal with this kind of network.

Agreement forests. A useful characterization of MinimumHybridization in terms of agreement forests was discovered by Baroni et al. [2], building on an idea in [20]. Bordewich and Semple used this characterization to show that Minimum-Hybridization is NP-hard. Such agreement forests play a fundamental role in this paper.

Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees. A partition $\mathcal{F} = \{\mathcal{L}_\rho, \mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k\}$ of $X \cup \{\rho\}$ is an *agreement forest* for $\mathcal{T}$ and $\mathcal{T}'$ if $\rho \in \mathcal{L}_\rho$ and the following conditions are satisfied:

(1) for all $i \in \{\rho, 1, 2, \ldots, k\}$, we have $\mathcal{T}|\mathcal{L}_i \cong \mathcal{T}'|\mathcal{L}_i$, and
(2) the trees in $\{\mathcal{T}(\mathcal{L}_i) : i \in \{\rho, 1, 2, \ldots, k\}\}$ and $\{\mathcal{T}'(\mathcal{L}_i) : i \in \{\rho, 1, 2, \ldots, k\}\}$ are vertex-disjoint subtrees of $\mathcal{T}$ and $\mathcal{T}'$, respectively.

In the definition above, the notation $\cong$ is used to denote a graph isomorphism that preserves leaf labels.

Note that even though an agreement forest is formally defined as a partition of the leaves, we often see the collection of trees $\{\mathcal{T}|\mathcal{L}_\rho, \mathcal{T}|\mathcal{L}_1, \ldots, \mathcal{T}|\mathcal{L}_k\}$ as the agreement forest. So, intuitively, an agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ can be seen as a collection of trees that can be obtained from either of $\mathcal{T}$ and $\mathcal{T}'$ by deleting a set of edges and subsequently "cleaning up" by deleting unlabeled leaves and suppressing indegree-1 outdegree-1 vertices (see Figure 2.1). Therefore, we often refer to the elements of an agreement forest as *components*.

The *size* of an agreement forest $\mathcal{F}$ is defined as its number of elements (components) and is denoted by $|\mathcal{F}|$.

A characterization of the hybridization number $h(\mathcal{T}, \mathcal{T}')$ in terms of agreement forests requires an additional condition. Let $\mathcal{F} = \{\mathcal{L}_\rho, \mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k\}$ be an agreement forest for $\mathcal{T}$ and $\mathcal{T}'$. Let $G_\mathcal{F}$ be the directed graph that has vertex set $\mathcal{F}$ and an edge $(\mathcal{L}_i, \mathcal{L}_j)$ if and only if $i \neq j$ and at least one of the two following conditions holds:

(1) the root of $\mathcal{T}(\mathcal{L}_i)$ is an ancestor of the root of $\mathcal{T}(\mathcal{L}_j)$ in $\mathcal{T}$;
(2) the root of $\mathcal{T}'(\mathcal{L}_i)$ is an ancestor of the root of $\mathcal{T}'(\mathcal{L}_j)$ in $\mathcal{T}'$.

The graph $G_\mathcal{F}$ is called the *inheritance graph* associated with $\mathcal{F}$. We call $\mathcal{F}$ an *acyclic agreement forest* for $\mathcal{T}$ and $\mathcal{T}'$ if $G_\mathcal{F}$ has no directed cycles. If $\mathcal{F}$ contains the smallest number of elements (components) over all acyclic agreement forests for $\mathcal{T}$ and $\mathcal{T}'$, we say that $\mathcal{F}$ is a *maximum acyclic agreement forest* for $\mathcal{T}$ and $\mathcal{T}'$. Note that such a forest is called a *maximum* acyclic agreement forest, even though one *minimizes* the number of elements, because in some sense the "agreement" is maximized. (Also note that acyclic agreement forests were called *good* agreement forests in [2].)

We define $m_a(\mathcal{T}, \mathcal{T}')$ to be the number of elements of a maximum acyclic agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ minus one. Also the problem of computing $m_a(\mathcal{T}, \mathcal{T}')$ has an optimization counterpart:

Problem: Maximum Acyclic Agreement Forest (MAAF).
Instance: Two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$.
Solution: An acyclic agreement forest $\mathcal{F}$ for $\mathcal{T}$ and $\mathcal{T}'$.
Objective: Minimize $|\mathcal{F}| - 1$.

We minimize $|\mathcal{F}| - 1$, rather than $|\mathcal{F}|$, following [8], because $|\mathcal{F}| - 1$ corresponds to the number of edges one needs to remove from either of the input trees to obtain $\mathcal{F}$

(after "cleaning up") and because of the relation we describe below between this problem and MinimumHybridization. Nevertheless, it can be shown that, from an approximation perspective, it does not matter whether one minimizes $|\mathcal{F}|$ or $|\mathcal{F}|-1$ (which is not obvious).

Theorem 2.1 (see [2, Theorem 2]). *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees. Then*

$$h(\mathcal{T},\mathcal{T}') = m_a(\mathcal{T},\mathcal{T}').$$

It is this characterization that was used by Bordewich and Semple [8] to show that MinimumHybridization is NP-hard. To show that also an approximation for one problem can be used to approximate the other problem, one needs the following slightly stronger result.

Theorem 2.2. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees. Then*

(i) *from a hybridization network $\mathcal{H}$ that displays $\mathcal{T}$ and $\mathcal{T}'$, one can construct in polynomial time an acyclic agreement forest $\mathcal{F}$ for $\mathcal{T}$ and $\mathcal{T}'$ such that $|\mathcal{F}|-1 \leq h(\mathcal{H})$; and*

(ii) *from an acyclic agreement forest $\mathcal{F}$ for $\mathcal{T}$ and $\mathcal{T}'$, one can construct in polynomial time a hybridization network $\mathcal{H}$ that displays $\mathcal{T}$ and $\mathcal{T}'$ such that $h(\mathcal{H}) \leq |\mathcal{F}|-1$.*

This result follows from the proof of [2, Theorem 2] using the earlier observation that we may assume that $\mathcal{H}$ is binary.

We now formally introduce the last optimization problem discussed in this paper. A *feedback vertex set* (FVS) of a directed graph D is a subset of the vertices that contains at least one vertex from each directed cycle in D. Equivalently, a subset V' of the vertices of D is an FVS if and only if removing V' from D gives a directed acyclic graph. The *minimum feedback vertex set problem on directed graphs* (DFVS) is defined as follows: Given a directed graph D, find an FVS of D that has minimum size.

Reductions and fixed parameter tractability. After establishing the NP-hardness of MinimumHybridization, the same authors showed that this problem is also FPT [7]. They show how to reduce a pair of rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$, such that the number of leaves of the reduced trees is bounded by $14h(\mathcal{T},\mathcal{T}')$, whence a brute-force algorithm can be used to solve the reduced instance, giving an FPT algorithm. A significantly faster FPT algorithm for MinimumHybridization uses a branching technique and has running time $O(3.18^r \cdot |X|)$ (or $O(3.18^r \cdot r + |X|^3)$ if it is used in combination with the mentioned kernelization) [40].

To describe the reductions, we need some additional definitions. Let $\mathcal{T}$ be a rooted binary phylogenetic X-tree. For $n \geq 2$, an *n-chain* of $\mathcal{T}$ is an n-tuple $(a_1, a_2, \ldots, a_n)$ of elements of $\mathcal{L}(\mathcal{T}) \setminus \{\rho\}$ such that the parent of a_1 is either the same as the parent of a_2 or the parent of a_1 is a child of the parent of a_2 and, for each $i \in \{2, 3, \ldots, n-1\}$, the parent of a_i is a child of the parent of a_{i+1}; i.e., the subgraph induced by $a_1, a_2, \ldots, a_n$ and their parents is a *caterpillar* (see Figure 2.1).

Now, let $A = (a_1, a_2, \ldots, a_n)$ be an n-chain that is common to two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$ with $n \geq 2$, and let $\mathcal{F}$ be an acyclic agreement forest for $\mathcal{T}$ and $\mathcal{T}'$. We say that A *survives* in $\mathcal{F}$ if there exists an element in $\mathcal{F}$ that is a superset of $\{a_1, a_2, \ldots, a_n\}$, while we say that A is *atomized* in $\mathcal{F}$ if each element in $\{a_1, a_2, \ldots, a_n\}$ is a singleton in $\mathcal{F}$ (see Figure 2.1). Furthermore, if T is a common pendant subtree of $\mathcal{T}$ and $\mathcal{T}'$, then we say that T *survives* in $\mathcal{F}$ if there is an element of $\mathcal{F}$ that is a superset of the label set of T.

The following lemma basically shows that we can reduce subtrees and chains. It

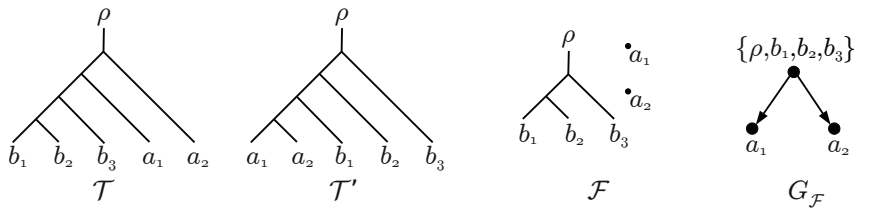

FIG. 2.1. *Two input trees $\mathcal{T}$ and $\mathcal{T}'$, an agreement forest $\mathcal{F}$ for $\mathcal{T}$ and $\mathcal{T}'$, and the inheritance graph $G_{\mathcal{F}}$. The trees have two common chains: (a_1, a_2) and (b_1, b_2, b_3). In the agreement forest $\mathcal{F}$, chain (a_1, a_2) is atomized while chain (b_1, b_2, b_3) survives. The agreement forest $\mathcal{F}$ is acyclic because $G_{\mathcal{F}}$ is acyclic.*

differs slightly from the corresponding lemma in [7] because we consider approximations, while Bordewich and Semple considered only optimal solutions in that paper.

LEMMA 2.3. *Let $\mathcal{F}$ be an acyclic agreement forest for two trees $\mathcal{T}$ and $\mathcal{T}'$. Then there exists an acyclic agreement forest $\mathcal{F}'$ for $\mathcal{T}$ and $\mathcal{T}'$ with $|\mathcal{F}'| \leq |\mathcal{F}|$ such that*

(i) *every common pendant subtree of $\mathcal{T}$ and $\mathcal{T}'$ survives in $\mathcal{F}'$; and*
(ii) *every common n-chain of $\mathcal{T}$ and $\mathcal{T}'$, with $n \geq 3$, either survives or is atomized in $\mathcal{F}'$.*

Moreover, $\mathcal{F}'$ can be obtained from $\mathcal{F}$ in polynomial time.

Proof. The proof follows from that of [7, Lemma 3.1]. There are two differences with [7, Lemma 3.1]. First, our result is slightly simpler because we consider two unweighted trees $\mathcal{T}$ and $\mathcal{T}'$, while the authors of [7] allow the unreduced trees $\mathcal{T}$ and $\mathcal{T}'$ to already have weights on 2-chains. Second, [7, Lemma 3.1] only shows the result for optimal agreement forests. However, a careful analysis of the proof of [7, Lemma 3.1] shows that it can also be used to prove this lemma. □

We are now ready to formally describe the aforementioned tree reductions. Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees, P a set that is initially empty, and $w : P \to \mathbb{Z}^+$ a weight function on the elements in P.

Subtree reduction. Replace any maximal pendant subtree with at least two leaves that is common to $\mathcal{T}$ and $\mathcal{T}'$ by a single leaf with a new label.

Chain reduction. Replace any maximal n-chain $(a_1, a_2, \ldots, a_n)$, with $n \geq 3$, that is common to $\mathcal{T}$ and $\mathcal{T}'$ by a 2-chain with new labels a and b. Moreover, add a new element (a, b) with weight $w(a, b) = n - 2$ to P.

Let $\mathcal{S}$ and $\mathcal{S}'$ be two rooted binary phylogenetic X'-trees that have been obtained from $\mathcal{T}$ and $\mathcal{T}'$ by first applying subtree reductions as often as possible and then applying chain reductions as often as possible. We call $\mathcal{S}$ and $\mathcal{S}'$ the *reduced tree pair* with respect to $\mathcal{T}$ and $\mathcal{T}'$. Note that a reduced tree pair always has an associated set P that contains one element for each chain reduction applied. Note that $\mathcal{S}$ and $\mathcal{S}'$ are unambiguously defined (up to the choice of the new labels) because maximal common pendant subtrees do not overlap and maximal common chains do not overlap. Moreover, applications of the chain reduction cannot create any new common pendant subtrees with at least two leaves. Hence, it is not necessary to apply subtree reductions again after the chain reductions.

Recall that every common n-chain, with $n \geq 3$, either survives or is atomized (Lemma 2.3). In $\mathcal{S}$ and $\mathcal{S}'$, such chains have been replaced by weighted 2-chains. Therefore, we are only interested in acyclic agreement forests for $\mathcal{S}$ and $\mathcal{S}'$ in which these weighted 2-chains either survive or are atomized. We therefore introduce a third notion of an agreement forest. Recall that P is the set of reduced (i.e., weighted) 2-chains. We say that an agreement forest $\mathcal{F}$ for $\mathcal{S}$ and $\mathcal{S}'$ is *legitimate* if it is acyclic

and every chain $(a, b) \in P$ either survives or is atomized in $\mathcal{F}$.

Let $\mathcal{F}$ be an agreement forest for $\mathcal{S}$ and $\mathcal{S}'$. The *weight* of $\mathcal{F}$, denoted by $w(\mathcal{F})$, is defined to be

$$w(\mathcal{F}) = |\mathcal{F}| - 1 + \sum_{(a,b)\in P:\ (a,b) \text{ is atomized in } \mathcal{F}} w(a, b).$$

Lastly, we define $f(\mathcal{S}, \mathcal{S}')$ to be the minimum weight of a legitimate agreement forest for $\mathcal{S}$ and $\mathcal{S}'$.

Then the following lemma says that computing the hybridization number of $\mathcal{T}$ and $\mathcal{T}'$ is equivalent to computing the minimum weight of a legitimate agreement forest for $\mathcal{S}$ and $\mathcal{S}'$. The second part of the lemma is necessary to show that an approximation to a reduced instance $\mathcal{S}$ and $\mathcal{S}'$ can be used to obtain an approximation to the original instance $\mathcal{T}$ and $\mathcal{T}'$.

LEMMA 2.4. *Let $\mathcal{T}$ and $\mathcal{T}'$ be a pair of rooted binary phylogenetic X-trees and let $\mathcal{S}$ and $\mathcal{S}'$ be the reduced tree pair with respect to $\mathcal{T}$ and $\mathcal{T}'$. Then*

(i) $h(\mathcal{S}, \mathcal{S}') \leq f(\mathcal{S}, \mathcal{S}') = h(\mathcal{T}, \mathcal{T}')$*; and*

(ii) *given a legitimate agreement forest $\mathcal{F}_S$ for $\mathcal{S}$ and $\mathcal{S}'$, we can find, in polynomial time, an acyclic agreement forest $\mathcal{F}$ for $\mathcal{T}$ and $\mathcal{T}'$ such that $|\mathcal{F}| - 1 = w(\mathcal{F}_S)$.*

Proof. In part (i), the inequality follows directly from the definition of f, while the equality is equivalent to [7, Proposition 3.2] if the unreduced trees $\mathcal{T}$ and $\mathcal{T}'$ are unweighted (i.e., if P is initially empty). Part (ii) follows from the proof of [7, Proposition 3.2]. ∎

The fixed parameter tractability of MINIMUMHYBRIDIZATION now follows from the next lemma, which bounds the number of leaves in a reduced tree pair.

LEMMA 2.5 (see [7, Lemma 3.3]). *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees, let $\mathcal{S}$ and $\mathcal{S}'$ be the reduced tree pair with respect to $\mathcal{T}$ and $\mathcal{T}'$, and let X' be the label set of $\mathcal{S}$ and $\mathcal{S}'$. If $h(\mathcal{T}, \mathcal{T}') > 0$, then $|X'| < 14h(\mathcal{T}, \mathcal{T}')$.*

We show in section 3 that the reduced trees have at most $9h(\mathcal{T}, \mathcal{T}')$ leaves. This improved bound will be important in the approximation-preserving reductions we give later in the paper.

3. An improved bound on the size of reduced instances of MINIMUMHYBRIDIZATION. We start with some definitions and an intermediate result. The bound on the size of the reduced instance will be proven in Theorem 3.2.

An *r-reticulation generator* (for short, *r-generator*) is defined to be a directed acyclic multigraph with a single vertex of indegree 0 and outdegree 1 (which we can think of as being labeled by ρ), precisely r reticulation vertices (indegree 2 and outdegree at most 1), and apart from that only vertices of indegree 1 and outdegree 2 [27]. The *sides* of an r-generator are its edges (the *edge sides*) and its vertices of indegree 2 and outdegree 0 (the *node sides*). The sides of a generator are the places where you can hang leaves, which we now formalize. Adding a set of labels L to an edge side (u, v) of an r-generator involves subdividing (u, v) into a path of $|L|$ internal vertices and, for each such internal vertex w, adding a new leaf w', an edge (w, w'), and labeling w' with some taxon from L (such that L bijectively labels the new leaves). On the other hand, adding a label l to a node side v consists of adding a new leaf y, an edge (v, y), and labeling y with l.

LEMMA 3.1. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees with no common pendant subtrees with at least two leaves, and let $\mathcal{H}$ be a hybridization network that displays $\mathcal{T}$ and $\mathcal{T}'$ with a minimum number of hybridization vertices. Then the*

network $\mathcal{H}'$ obtained from $\mathcal{H}$ by deleting all $|X|$ leaves and suppressing each resulting vertex v with $d^+(v) = d^-(v) = 1$ is an $h(\mathcal{H})$-generator.

Proof. By construction, $\mathcal{H}'$ contains the same number of hybridization vertices as $\mathcal{H}$. Additionally, by the definition of a binary hybridization network, no vertex has indegree 2 and outdegree greater than 1, indegree greater than 2, or indegree and outdegree both 1. Now, we claim that $\mathcal{H}'$ does not have any vertex with indegree 1 and outdegree 0. To see that this holds, suppose that there exists a vertex v in $\mathcal{H}'$ such that $d^-(v) = 1$ and $d^+(v) = 0$. Then v has two children in $\mathcal{H}$. Since $d^+(v) = 0$ in $\mathcal{H}'$, no hybridization vertex can be reached by a directed path from v in $\mathcal{H}$. This means that the subnetwork of $\mathcal{H}$ rooted at v is actually a rooted tree, contradicting the fact that $\mathcal{T}$ and $\mathcal{T}'$ do not have any common pendant subtree with two or more leaves. We may thus conclude that $\mathcal{H}'$ conforms to the definition of an $h(\mathcal{H})$-generator. ∎

Conversely, by inverting the operations of suppression and deletion, $\mathcal{H}$ can be obtained from the $h(\mathcal{H})$-generator $\mathcal{H}'$ associated with $\mathcal{H}$ by adding leaves to its sides (in the sense described at the start of this section). This relies on the intuitive fact that, modulo leaves and suppression, the $h(\mathcal{H})$-generator obtained in Lemma 3.1 has essentially the same topology as $\mathcal{H}$. A similar technique was described in [27] in a somewhat different context.

THEOREM 3.2. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees, and let $\mathcal{S}$ and $\mathcal{S}'$ be the reduced tree pair on X' with respect to $\mathcal{T}$ and $\mathcal{T}'$. If $h(\mathcal{T}, \mathcal{T}') > 0$, then $|X'| < 9h(\mathcal{T}, \mathcal{T}')$.*

Proof. Let $\mathcal{H}'$ be the $h(\mathcal{H})$-generator that is associated with a hybridization network $\mathcal{H}$ for $\mathcal{S}$ and $\mathcal{S}'$ whose number of hybridization vertices is minimized, i.e., $h(\mathcal{H}) = h(\mathcal{S}, \mathcal{S}')$. By definition, $\mathcal{H}'$ has the following vertices:

- $r = h(\mathcal{H})$ reticulations; in particular r_0 reticulations with indegree 2 and outdegree 0 and r_1 reticulations with indegree 2 and outdegree 1;
- s vertices with indegree 1 and outdegree 2; and
- one vertex labeled ρ with indegree 0 and outdegree 1.

The total indegree of $\mathcal{H}'$ is $2r_0 + 2r_1 + s$. The total outdegree of $\mathcal{H}'$ is $r_1 + 2s + 1$. Hence, $2r_0 + 2r_1 + s = r_1 + 2s + 1$, implying $s = 2r_0 + r_1 - 1$. Moreover, the total number of edges of $\mathcal{H}'$, $|E(\mathcal{H}')|$, equals the total indegree and, therefore,

$$|E(\mathcal{H}')| = 2r_0 + 2r_1 + s = 2r_0 + 2r_1 + 2r_0 + r_1 - 1 = 4r_0 + 3r_1 - 1. \tag{3.1}$$

Note that for each of the r_0 node sides v in $\mathcal{H}'$ the child of v in $\mathcal{H}$ is a single leaf (because otherwise there would be a common pendant subtree with at least two leaves). Moreover, each edge side in $\mathcal{H}'$ cannot correspond to a directed path in $\mathcal{H}$ that consists of more than three edges since, otherwise, $\mathcal{S}$ and $\mathcal{S}'$ would have a common n-chain, with $n \geq 3$. Thus, $\mathcal{H}$ can have at most two leaves per edge side of $\mathcal{H}'$ and one leaf per node side of $\mathcal{H}'$. Thus, the total number of leaves $|X'|$ of $\mathcal{H}$ is bounded by

$$\begin{aligned} |X'| &\leq 2|E(\mathcal{H}')| + r_0 \\ &= 2(4r_0 + 3r_1 - 1) + r_0 \\ &= 9r_0 + 6r_1 - 2 \\ &\leq 9r - 2 \\ &< 9h(\mathcal{S}, \mathcal{S}') \\ &\leq 9h(\mathcal{T}, \mathcal{T}'), \end{aligned}$$

where the last inequality follows from Lemma 2.4. ∎

4. An approximation-preserving reduction from MinimumHybridization to DFVS. We start by proving the following theorem, which refers to wDFVS, the *weighted* variant of DFVS where every vertex is attributed a weight and the weight of an FVS is simply the sum of the weights of its constituent vertices. Later in the section we will prove a corresponding result for DFVS.

Theorem 4.1. *If, for some $c \geq 1$, there exists a polynomial-time c-approximation for* wDFVS*, then there exists a polynomial-time* $6c$*-approximation for* Minimum-Hybridization*.*

Throughout this section, let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees, and let $\mathcal{S}$ and $\mathcal{S}'$ be the reduced tree pair on X' with respect to $\mathcal{T}$ and $\mathcal{T}'$. Using Lemma 2.3, we assume throughout this section without loss of generality that $\mathcal{T}$ and $\mathcal{T}'$ do not contain any common pendant subtrees with at least two leaves. Thus, the reduced tree pair $\mathcal{S}$ and $\mathcal{S}'$ can be obtained from $\mathcal{T}$ and $\mathcal{T}'$ by applying the chain reduction only.

Before starting the proof, we need some additional definitions and lemmas. We say that a common chain (a, b) of $\mathcal{S}$ and $\mathcal{S}'$ is a *reduced chain* if it is not a common chain of $\mathcal{T}$ an $\mathcal{T}'$. Otherwise, (a, b) is an *unreduced chain*. Furthermore, a taxon $\ell \in X' \cup \{\rho\}$, is a *nonchain taxon* if it does not label a leaf of a reduced or unreduced chain of $\mathcal{S}$ and $\mathcal{S}'$. Now, let $\mathcal{B}_\mathcal{S}$ be the forest that exactly contains the following elements:

1. for each nonchain taxon ℓ of $\mathcal{S}$ and $\mathcal{S}'$, a *nonchain element* $\{\ell\}$; and
2. for each reduced and unreduced chain (a, b) of $\mathcal{S}$ and $\mathcal{S}'$, an element $\{a, b\}$.

Clearly, $\mathcal{B}_\mathcal{S}$ is an agreement forest for $\mathcal{S}$ and $\mathcal{S}'$, and we refer to it as a *chain forest* for $\mathcal{S}$ and $\mathcal{S}'$. Now, obtain $\mathcal{B}_\mathcal{T}$ from $\mathcal{B}_\mathcal{S}$ by replacing each element in $\mathcal{B}_\mathcal{S}$ that contains two labels of a reduced chain, say (a, b), of $\mathcal{S}$ and $\mathcal{S}'$ with the label set that precisely contains all labels of the common n-chain that has been reduced to (a, b) in the course of obtaining $\mathcal{S}$ and $\mathcal{S}'$ from $\mathcal{T}$ and $\mathcal{T}'$, respectively. The set $\mathcal{B}_\mathcal{T}$ is an agreement forest for $\mathcal{T}$ and $\mathcal{T}'$, and we refer to it as a *chain forest* for $\mathcal{T}$ and $\mathcal{T}'$. Since the chain reduction can be performed in polynomial time [7], the chain forests $\mathcal{B}_\mathcal{S}$ and $\mathcal{B}_\mathcal{T}$ can also be calculated in polynomial time from $\mathcal{T}$ and $\mathcal{T}'$. Lastly, each element in $\mathcal{B}_\mathcal{T}$ whose members label the leaves of a common n-chain in $\mathcal{T}$ and $\mathcal{T}'$ with $n \geq 2$ is referred to as a *chain element*.

The next lemma bounds the number of elements in a chain forest.

Lemma 4.2. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees. Let $\mathcal{S}$ and $\mathcal{S}'$ be the reduced tree pair with respect to $\mathcal{T}$ and $\mathcal{T}'$. Furthermore, let $\mathcal{B}_\mathcal{S}$ and $\mathcal{B}_\mathcal{T}$ be the chain forests for $\mathcal{S}$ and $\mathcal{S}'$ and for $\mathcal{T}$ and $\mathcal{T}'$, respectively. Then $|\mathcal{B}_\mathcal{T}| = |\mathcal{B}_\mathcal{S}| < 5h(\mathcal{T}, \mathcal{T}')$.*

Proof. By construction of $\mathcal{B}_\mathcal{T}$ from $\mathcal{B}_\mathcal{S}$, it immediately follows that $|\mathcal{B}_\mathcal{T}| = |\mathcal{B}_\mathcal{S}|$. To show that $|\mathcal{B}_\mathcal{S}| < 5h(\mathcal{T}, \mathcal{T}')$ let $\mathcal{H}$ be a hybridization network that displays $\mathcal{S}$ and $\mathcal{S}'$ and such that its number of hybridization vertices is minimized over all such networks. Furthermore, let $\mathcal{H}'$ be the $h(\mathcal{H})$-generator associated with $\mathcal{H}$. As in the proof of Theorem 3.2, let r_0 be the number of node sides, i.e., reticulations with indegree 2 and outdegree 0, in $\mathcal{H}'$, and let r_1 be the number of reticulations in $\mathcal{H}'$ with indegree 2 and outdegree 1. Again, $r_0 + r_1 = h(\mathcal{H}') = h(\mathcal{S}, \mathcal{S}')$. Recall that, to obtain $\mathcal{H}$ from $\mathcal{H}'$, we add one leaf to each node side of $\mathcal{H}'$, corresponding to a singleton in $\mathcal{B}_\mathcal{S}$, and at most two leaves to each edge side of $\mathcal{H}'$. Each edge side of $\mathcal{H}'$ to which we add two taxa corresponds to a 2-chain of $\mathcal{S}$ and $\mathcal{S}'$ and, therefore, to a single element in $\mathcal{B}_\mathcal{S}$. Hence, using (3.1) and Lemma 2.4, we have

$$|\mathcal{B}_\mathcal{T}| = |\mathcal{B}_\mathcal{S}| \leq |E(\mathcal{H}')| + r_0 = 5r_0 + 3r_1 - 1 < 5(r_0 + r_1) = 5h(\mathcal{S}, \mathcal{S}') \leq 5h(\mathcal{T}, \mathcal{T}'). \qquad \square$$

Consider again the chain forest $\mathcal{B}_\mathcal{T}$ for $\mathcal{T}$ and $\mathcal{T}'$. We define a *$\mathcal{B}_\mathcal{T}$-splitting* as an acyclic agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ that can be obtained from $\mathcal{B}_\mathcal{T}$ by repeated replacements of a chain element $\{a_1, a_2, \ldots, a_n\}$ with the elements $\{a_1\}, \{a_2\}, \ldots, \{a_n\}$.

LEMMA 4.3. *Let $\mathcal{B}_\mathcal{T}$ be the chain forest for two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$. Let $\{a_1, a_2, \ldots, a_n\}$ be a chain element in $\mathcal{B}_\mathcal{T}$, and let $\mathcal{L}_j$ be a nonchain element in $\mathcal{B}_\mathcal{T}$. Furthermore, let $\mathcal{B}'_\mathcal{T} = (\mathcal{B}_\mathcal{T} - \{\{a_1, a_2, \ldots, a_n\}\}) \cup \{\{a_1\}, \{a_2\}, \ldots, \{a_n\}\}$. Then*

(i) *no directed cycle of $G_{\mathcal{B}'_\mathcal{T}}$ passes through an element of $\{\{a_1\}, \{a_2\}, \ldots, \{a_n\}\}$; and*

(ii) *no directed cycle of $G_{\mathcal{B}_\mathcal{T}}$ passes through $\mathcal{L}_j$.*

Proof. By the definition of $\mathcal{B}_\mathcal{T}$, note that $|\mathcal{L}_j| = 1$. If $\mathcal{L}_j = \{\rho\}$, then the indegree of $\mathcal{L}_j$ is 0 in $G_{\mathcal{B}_\mathcal{T}}$. Otherwise, if $\mathcal{L}_j \neq \{\rho\}$, then its element labels a leaf of $\mathcal{T}$ and $\mathcal{T}'$, and thus the outdegree of $\mathcal{L}_j$ is 0 in $G_{\mathcal{B}_\mathcal{T}}$. Furthermore, since each element in $\{\{a_1\}, \{a_2\}, \ldots, \{a_n\}\}$ also labels a leaf of $\mathcal{T}$ and $\mathcal{T}'$, the outdegree of the vertices $a_1, a_2, \ldots, a_n$ in $G_{\mathcal{B}'_\mathcal{T}}$ is 0. This establishes the lemma. ∎

Let OPT($\mathcal{B}_\mathcal{T}$-splitting) denote the size of a $\mathcal{B}_\mathcal{T}$-splitting of smallest size.

LEMMA 4.4. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees, and let $\mathcal{B}_\mathcal{T}$ be the chain forest for $\mathcal{T}$ and $\mathcal{T}'$. Then* OPT($\mathcal{B}_\mathcal{T}$-splitting) $< 6h(\mathcal{T}, \mathcal{T}')$.

Proof. Let $\mathcal{F}_\mathcal{T}$ be a maximum acyclic agreement forest for $\mathcal{T}$ and $\mathcal{T}'$. By Lemma 2.3, we may assume that, for $n \geq 3$, every common n-chain of $\mathcal{T}$ and $\mathcal{T}'$ survives or is atomized in $\mathcal{F}_\mathcal{T}$. In this proof, we see an agreement forest as a collection of trees (see the remark below the definition in section 2). Thus, $\mathcal{F}_\mathcal{T}$ can be obtained from $\mathcal{T}$ (or equivalently from $\mathcal{T}'$) by deleting an $(|\mathcal{F}_\mathcal{T}| - 1)$-sized subset, say $E_{\mathcal{F}_\mathcal{T}}$, of the edges of $\mathcal{T}$ and cleaning up. Similarly, $\mathcal{B}_\mathcal{T}$ can be obtained from $\mathcal{T}$ (or equivalently from $\mathcal{T}'$) by deleting a $(|\mathcal{B}_\mathcal{T}| - 1)$-sized subset, say $E_{\mathcal{B}_\mathcal{T}}$, and cleaning up. Now consider the forest $\mathcal{B}'_\mathcal{T}$ obtained from $\mathcal{T}$ by removing the edge set $E_{\mathcal{F}_\mathcal{T}} \cup E_{\mathcal{B}_\mathcal{T}}$ and cleaning up.

We claim that $\mathcal{B}'_\mathcal{T}$ is a $\mathcal{B}_\mathcal{T}$-splitting. To see this, first observe that $\mathcal{B}'_\mathcal{T}$ is an acyclic agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ because it can be obtained by removing the edge set $E_{\mathcal{B}_\mathcal{T}}$ from $\mathcal{F}_\mathcal{T}$ and cleaning up. Hence, to show that $\mathcal{B}'_\mathcal{T}$ is a $\mathcal{B}_\mathcal{T}$-splitting, it is left to show that it can be obtained from $\mathcal{B}_\mathcal{T}$ by repeated replacements of a caterpillar on $\{a_1, a_2, \ldots, a_n\}$ by isolated vertices $\{a_1\}, \{a_2\}, \ldots, \{a_n\}$. By its definition, $\mathcal{B}'_\mathcal{T}$ can be obtained from $\mathcal{B}_\mathcal{T}$ by removing edges and cleaning up. Thus, what is left to prove is that each chain either survives or is atomized. Since by assumption all n-chains with $n \geq 3$ either survive or are atomized in $\mathcal{F}_\mathcal{T}$, the same holds for $\mathcal{B}'_\mathcal{T}$. Now observe that, by the definition of $\mathcal{B}_\mathcal{T}$, each 2-chain is a component on its own in $\mathcal{B}_\mathcal{T}$. Since $\mathcal{B}'_\mathcal{T}$ can be obtained by removing edges from $\mathcal{B}_\mathcal{T}$, it follows that each 2-chain either survives or is atomized in $\mathcal{B}'_\mathcal{T}$.

As the size of $\mathcal{B}'_\mathcal{T}$ is equal to the number of edges removed to obtain it from $\mathcal{T}$ plus one, we have

$$|\mathcal{B}'_\mathcal{T}| \leq |E_{\mathcal{F}_\mathcal{T}}| + |E_{\mathcal{B}_\mathcal{T}}| + 1 = |\mathcal{F}_\mathcal{T}| - 1 + |\mathcal{B}_\mathcal{T}| < h(\mathcal{T}, \mathcal{T}') + 5h(\mathcal{T}, \mathcal{T}') = 6h(\mathcal{T}, \mathcal{T}'),$$

where Lemma 4.2 is used to bound $|\mathcal{B}_\mathcal{T}|$. This establishes the lemma. ∎

We are now in a position to prove the main result of this section.

Proof of Theorem 4.1. Throughout this proof, let $n \geq 2$. Furthermore, let $\mathcal{B}_\mathcal{T}$ be the chain forest for $\mathcal{T}$ and $\mathcal{T}'$, and let G be the graph obtained from the inheritance graph $G_{\mathcal{B}_\mathcal{T}}$ by subsequently

1. weighting each vertex that corresponds to a common n-chain $(a_1, a_2, \ldots, a_n)$ of $\mathcal{T}$ and $\mathcal{T}'$ with weight n;

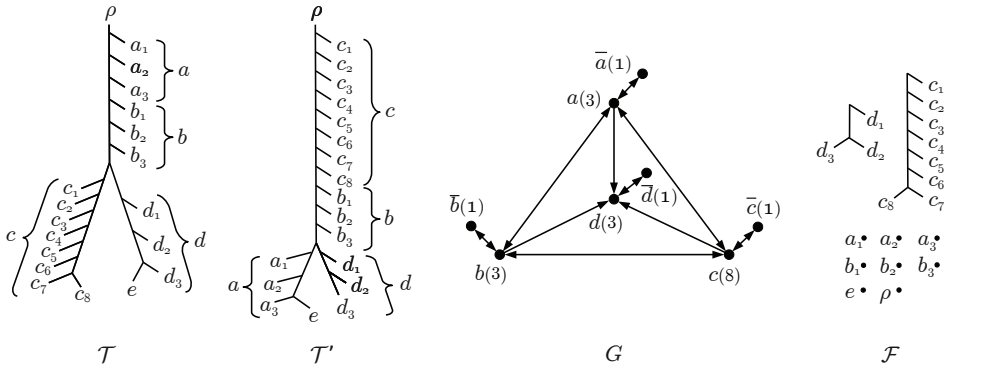

FIG. 4.1. *Two input trees $\mathcal{T}$ and $\mathcal{T}'$, their graph G (with weights between parentheses), and an acyclic agreement forest $\mathcal{F}$ of $\mathcal{T}$ and $\mathcal{T}'$. Note that $\mathcal{F}$ is a $\mathcal{B}_{\mathcal{T}}$-splitting because it can be obtained from the chain forest $\mathcal{B}_{\mathcal{T}}$ by atomizing chains $a = (a_1, \ldots, a_3)$ and $b = (b_1, \ldots, b_3)$. Also note that $\mathcal{F}$ has 10 components, which is equal to the weight of a minimum FVS $\{a, b, \bar{c}, \bar{d}\}$ of G, 8, plus two nonchain taxa (ρ and e).*

2. deleting each vertex that corresponds to a nonchain taxon in $\mathcal{B}_{\mathcal{T}}$; and
3. for each remaining vertex v, creating a new vertex $\bar{v}$ with weight 1 and two new edges $(v, \bar{v})$ and $(\bar{v}, v)$.

Let w be the weight function on the vertices of G defined this way. See Figure 4.1 for an example of the construction of G. We call the added vertices $\bar{v}$ the *barred* vertices of G. Note that each common n-chain of $\mathcal{T}$ and $\mathcal{T}'$ is represented by a vertex and its barred vertex in G. As $\mathcal{B}_{\mathcal{T}}$ can be calculated in polynomial time, the construction of G also takes polynomial time, and the size of G is clearly polynomial in the cardinality of $\mathcal{B}_{\mathcal{T}}$.

Now, regarding G as an instance of wDFVS, we claim the following.

Claim. There exists a $\mathcal{B}_{\mathcal{T}}$-splitting of size $k+s$, where s is the number of nonchain elements in $\mathcal{B}_{\mathcal{T}}$ if and only if G has an FVS of weight k.

Suppose that $\mathcal{B}'_{\mathcal{T}}$ is a $\mathcal{B}_{\mathcal{T}}$-splitting of size $k + s$. Hence, k is equal to the number of chain elements in $\mathcal{B}_{\mathcal{T}}$ that are also elements in $\mathcal{B}'_{\mathcal{T}}$ plus the total number of leaves in common n-chains that are atomized in $\mathcal{B}'_{\mathcal{T}}$. Let $\bar{\mathcal{B}}'_{\mathcal{T}}$ be the forest that has been obtained from $\mathcal{B}'_{\mathcal{T}}$ by deleting all singletons, and let $G_{\bar{\mathcal{B}}'_{\mathcal{T}}}$ be its inheritance graph. Since $G_{\mathcal{B}'_{\mathcal{T}}}$ is acyclic, $G_{\bar{\mathcal{B}}'_{\mathcal{T}}}$ is also acyclic. Now, let G' be the directed graph that has been obtained from G in the following way. For each nonbarred vertex v in G, delete v if v corresponds to an n-chain of $\mathcal{T}$ and $\mathcal{T}'$ that is atomized in $\mathcal{B}'_{\mathcal{T}}$, and delete $\bar{v}$ if v corresponds to an n-chain of $\mathcal{T}$ and $\mathcal{T}'$ that is not atomized in $\mathcal{B}'_{\mathcal{T}}$. Note that for each 2-cycle $(v, \bar{v}, v)$ of G either v or $\bar{v}$ is not a vertex of G' because each n-chain that is common to $\mathcal{T}$ and $\mathcal{T}'$ is either atomized or not in $\mathcal{B}'_{\mathcal{T}}$. This in turn implies that G' is acyclic because $G_{\bar{\mathcal{B}}'_{\mathcal{T}}}$ is isomorphic to $G' \backslash \bar{V}$, where $\bar{V}$ precisely contains all barred vertices of G'. Hence, an FVS of G, say V, contains each vertex of G that is not a vertex of G'. Furthermore, by the weighting of G, it follows that the weight of V is exactly k.

Conversely, suppose that there exists an FVS of G, say V, with weight k. This implies that we can remove a set V_1 of barred vertices and a set $V_2 = V \backslash V_1$ of nonbarred vertices such that $\sum_{v_i \in V_2} w(v_i) + |V_1| = k$ and the graph $G' = G \backslash V$ is acyclic. For each vertex $v_i \in V_2$, let $A_i = (a_{i,1}, a_{i,2}, \ldots, a_{i,n})$ be its associated common chain of $\mathcal{T}$ and $\mathcal{T}'$, and let $w(v_i)$ be the number of elements in A_i. Furthermore, let V'_1

be the subset of V_1 that contains precisely each vertex $\bar{v}$ of V_1 for which $v \notin V_2$. If $|V_1'| < |V_1|$, then it is easily checked that $V_1' \cup V_2$ is an FVS of G whose weight is strictly less than k. Therefore, we may assume for the remainder of this proof that $|V_1'| = |V_1|$. Now, let $\mathcal{B}'_{\mathcal{T}}$ be the forest that has been obtained from $\mathcal{B}_{\mathcal{T}}$ in the following way. For each vertex v_i in V_2, replace A_i in $\mathcal{B}_{\mathcal{T}}$ with the elements $\{a_{i,1}\}, \{a_{i,2}\}, \ldots, \{a_{i,n}\}$. Thus, A_i is atomized in $\mathcal{B}'_{\mathcal{T}}$. We next construct the inheritance graph $G_{\mathcal{B}'_{\mathcal{T}}}$ from $G_{\mathcal{B}_{\mathcal{T}}}$. For each vertex v of $G_{\mathcal{B}_{\mathcal{T}}}$ that corresponds to a common n-chain $(a_1, a_2, \ldots, a_n)$ of $\mathcal{T}$ and $\mathcal{T}'$ that is atomized in $\mathcal{B}'_{\mathcal{T}}$, replace v with the vertices $a_1, a_2, \ldots, a_n$, delete each edge (v, w) of $G_{\mathcal{B}_{\mathcal{T}}}$, and replace each edge (u, v) of $G_{\mathcal{B}_{\mathcal{T}}}$ with the edges $(u, a_1), (u, a_2), \ldots, (u, a_n)$. By Lemma 4.3, the vertices $a_1, a_2, \ldots, a_n$ have outdegree 0 in $G_{\mathcal{B}'_{\mathcal{T}}}$. Noting that there is a natural bijection between the cycles in $G_{\mathcal{B}_{\mathcal{T}}}$ and the cycles in G that do not pass through any barred vertex, it follows that, as G' is acyclic, $G_{\mathcal{B}'_{\mathcal{T}}}$ is also acyclic. Hence, $\mathcal{B}'_{\mathcal{T}}$ is a $\mathcal{B}_{\mathcal{T}}$-splitting for $\mathcal{T}$ and $\mathcal{T}'$. The claim now follows from

$$|\mathcal{B}'_{\mathcal{T}}| = s + \sum_{v_i \in V_2} w(v_i) + |V_1| = s + k.$$

It remains to show that the reduction is approximation preserving. Suppose that there exists a polynomial-time c-approximation for wDFVS. Let k be the weight of a solution returned by this algorithm, and let k^* be the weight of an optimal solution. By the above claim, we can then construct a solution to MAAF of size $k + s$, from which we can obtain a solution to MinimumHybridization with value $k + s - 1$ by Theorem 2.2. We have

$$k + s - 1 < ck^* + s \leq ck^* + cs = c(k^* + s) = c \cdot \mathrm{OPT}(\mathcal{B}_{\mathcal{T}}\text{-splitting})$$

and, thus, a constant factor c-approximation for finding an optimal $\mathcal{B}_T$-splitting. Now, by Lemma 4.4,

$$k + s - 1 \leq c \cdot \mathrm{OPT}(\mathcal{B}_{\mathcal{T}}\text{-splitting}) \leq 6c \cdot h(\mathcal{T}, \mathcal{T}'),$$

thereby establishing that if there exists a polynomial-time c-approximation for wDFVS, then there exists a polynomial-time $6c$-approximation for Minimum-Hybridization. This concludes the proof of the theorem. ∎

It is not too difficult to extend Theorem 4.1 to DFVS, i.e., the unweighted variant of the problem.

Theorem 4.5. *If, for some $c \geq 1$, there exists a polynomial-time c-approximation for* DFVS*, then there exists a polynomial-time $6c$-approximation for* Minimum-Hybridization.

Proof. In the proof of Theorem 4.1 we created an instance G of wDFVS. Let w be the weight function on the vertices of G. Note that the function is nonnegative and integral and for every vertex $v \in G$, $w(v) \leq |X|$, i.e., the weight function is polynomially bounded in the input size. We create an instance G' of DFVS as follows. For each vertex v in G we create $w(v)$ vertices in G' $v_1, \ldots, v_{w(v)}$. For each edge (u, v) in G we introduce edges $\{(u_i, v_j) | 1 \leq i \leq w(u), 1 \leq j \leq w(v)\}$ in G'. Solutions to wDFVS(G) and DFVS(G') are very closely related, which allows us to construct in polynomial-time a c-approximation algorithm for wDFVS from a c-approximation for DFVS. Formally, what we will demonstrate is an L-reduction [32] from wDFVS to DFVS with coefficients $\alpha = \beta = 1$ which works for instances with polynomially bounded weights. Specifically, consider any FVS F' of G' of size k. We create an

FVS F of G as follows. For each vertex $v \in G$, we include v in F if and only if *all* the vertices $v_1, \ldots, v_{w(v)}$ are in F'. Note that the weight of F is less than or equal to k. To see that F is an FVS, suppose some cycle $C = u, v, w, \ldots, u$ survives in G. But then, for each vertex $u \in C$, some vertex u_i survives in G', which means a cycle also survived in G', contradicting the assumption that F' is an FVS. In the other direction, observe that any weight k FVS F of G can be transformed into an FVS F' of G' with size k as follows: for each $v \in F$, place all $v_1, \ldots, v_{w(v)}$ in F'. □

Notice that Theorem 4.1 does not hold only for constant c, an observation used in the next corollary.

COROLLARY 4.6. *There exists a polynomial-time* $O(\log r \log\log r)$*-approximation for* MINIMUMHYBRIDIZATION*, where* $r = h(\mathcal{T}, \mathcal{T}')$

Proof. In [15], which extended [37], a polynomial-time approximation algorithm for wDFVS is presented whose approximation ratio is $O(\min(\log |V| \log\log |V|, \log \tau^* \log\log \tau^*))$, where $|V|$ is the number of vertices in the wDFVS instance and τ^* is the optimal *fractional* solution value of the problem. We show that in the wDFVS instance G that we create in the proof of Theorem 4.1, both the number of vertices in G and the weight of the optimal fractional solution value of wDFVS(G) are $O(r)$. To see that G has at most $O(r)$ vertices, observe that G contains two vertices for every chain element in the chain forest $\mathcal{B}_\mathcal{T}$, and that (by Lemma 4.2) $|\mathcal{B}_\mathcal{T}| < 5r$. Second, recall from Lemma 4.4 that OPT($\mathcal{B}_\mathcal{T}$-splitting) $< 6r$. By construction, OPT($\mathcal{B}_\mathcal{T}$-splitting) is an upper bound on the optimum solution value of wDFVS(G), hence on τ^*. Thus, given G as input, the algorithm in [15] constructs an FVS that is at most a factor $O(\log r \log\log r)$ larger than the true optimal solution of wDFVS(G). As shown in the proof of Theorem 4.1 this can be used to obtain an approximation ratio at most six times larger for MAAF, which is clearly also $O(\log r \log\log r)$. □

Finally, note that for a given instance the actual approximation ratio obtained by Corollary 4.6 will sometimes be determined by $|V|$, and sometimes by τ^*, and can potentially be significantly smaller than $O(\log r \log\log r)$. For example, if there are very few chains in the chain forest, but they are all extremely long, then it can happen that $|V| \ll \tau^*$. Conversely, if the chain forest contains many short chains, and only a small number of them need to be atomized to attain acyclicity, then it can happen that $\tau^* \ll |V|$.

5. An approximation-preserving reduction from DFVS to MINIMUM-HYBRIDIZATION. In this section we prove the following theorem.

THEOREM 5.1. *If, for some constant* $c \geq 1$*, there exists a polynomial-time* c*-approximation algorithm for* MINIMUMHYBRIDIZATION*, then there exists a polynomial-time* $(c + \epsilon)$*-approximation algorithm for* DFVS *for all* $\epsilon > 0$.

Proof. We show an approximation preserving reduction from DFVS to MAAF. The theorem then follows because of the equivalence of MAAF and MINIMUM-HYBRIDIZATION described in Theorem 2.2.

Let $D = (V, A)$ be an instance of DFVS. First we transform D into an auxiliary graph $D' = (V', A')$. For a vertex v of D, we denote the parents of v as $u_1, u_2, \ldots, u_{d^-(v)}$ and the children of v as $w_1, w_2, \ldots, w_{d^+(v)}$. (To facilitate the exposition, we assume a total order on the parents of each vertex and on the children of each vertex.) We construct the graph D' as follows. For every vertex $v \in V$, D' has vertices $v_{\text{in}}^{u_1}, v_{\text{in}}^{u_2}, \ldots, v_{\text{in}}^{u_{d^-(v)}}$, vertices v^- and v^+, as well as vertices $v_{\text{out}}^{w_1}, v_{\text{out}}^{w_2}, \ldots, v_{\text{out}}^{w_{d^+(v)}}$. The edges of D' are as follows. For each vertex $v \in V$, D' has edges from each of $v_{\text{in}}^{u_1}, v_{\text{in}}^{u_2}, \ldots, v_{\text{in}}^{u_{d^-(v)}}$ to v^-, an edge from v^- to v^+, and edges from v^+ to each of

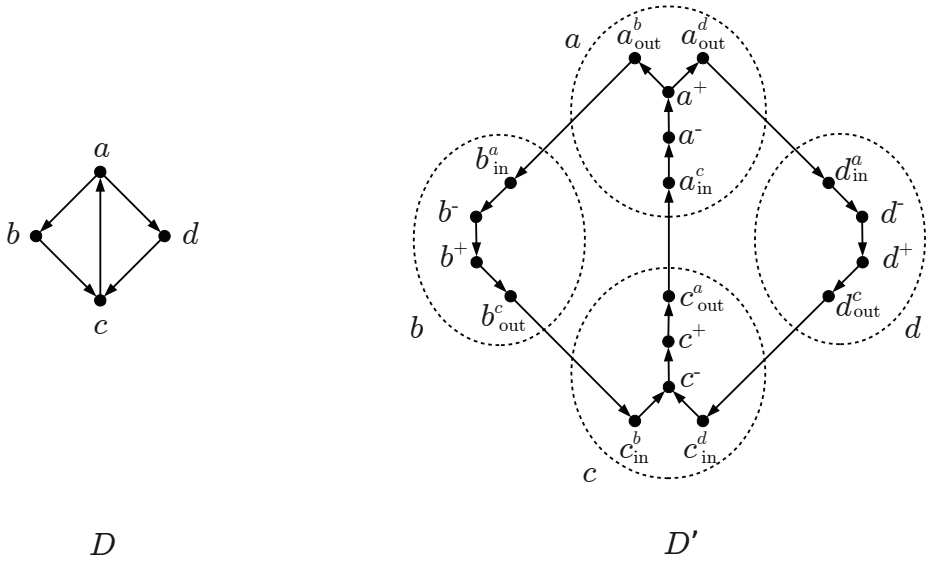

FIG. 5.1. *An instance D of DFVS and the modified graph D'.*

$v_{\text{out}}^{w_1}, v_{\text{out}}^{w_2}, \ldots, v_{\text{out}}^{w_{d^+(v)}}$. In addition, for each edge (u, v) of D, there is an edge $(u^v_{\text{out}}, v^u_{\text{in}})$ in D'. This concludes the construction of D'. An example is given in Figure 5.1.

We now first show that D has an FVS of size at most f if and only if D' has an FVS of size at most f. Observe that each directed cycle of D corresponds to a directed cycle of D', and vice versa. Also notice that any cycle in D' that contains a vertex v^u_{in} or a vertex v^w_{out} also contains the vertices v^- and v^+. Hence, for any FVS F of D' we can create an FVS F' of D' of at most the same size and containing only vertices of type v^-. We assume from now on that any FVS of D' is of this form. It is now obvious that F is an FVS of D if and only if $F' = \{v^- \in V' \mid v \in F\}$ is an FVS of D'.

Intuitively, the idea of our reduction is as follows. We will construct two rooted binary trees $\mathcal{T}$ and $\mathcal{T}'$ consisting of long chains. We build them in such a way that the graph D' is basically the inheritance graph of the chain forest for $\mathcal{T}$ and $\mathcal{T}'$. This graph can be made acyclic by atomizing some of the chains. Thus, solving DFVS on D' is basically equivalent to deciding which chains to atomize. We make all the chains that can be atomized of the same length. Hence, since each chain that is atomized adds the same number of components to the agreement forest, solving DFVS on D' is essentially equivalent to finding a maximum acyclic agreement forest for $\mathcal{T}$ and $\mathcal{T}'$.

Before we proceed, we need some more definitions. Recall that an *n-chain* of a tree is an n-tuple $(a_1, a_2, \ldots, a_n)$ of leaves such that either the parent of a_1 is the same as the parent of a_2 or the parent of a_1 is a child of the parent of a_2 and, for each $i \in \{2, 3, \ldots, n-1\}$, the parent of a_i is a child of the parent of a_{i+1}. A tree T whose leaf set $\mathcal{L}(T)$ is a chain of T is called a *caterpillar* on $\mathcal{L}(T)$.[1] It is easy to see that, for every chain C, there exists a unique caterpillar on C. By *hanging* a chain C below a leaf x, we mean the following: subdivide the edge entering x by a new vertex v and add an edge from v to the root of the caterpillar on C. When we hang a chain C_1

[1] In this context a caterpillar does *not* have an extra vertex labeled ρ.

below a chain C_2, we hang the caterpillar on C_1 below the lowest leaf (or a lowest leaf) x_1 of C_2. By *replacing* a leaf x by a chain C we mean deleting x and adding an edge from its former parent to the root of the caterpillar on C.

We are now ready to construct an instance of MAAF. The trees, $\mathcal{T}$ and $\mathcal{T}'$, will be built of chains of three types: x-type, y-type, and z-type. The x-type chains have length ℓ, while the y-type and z-type chains have length L (with $L \gg \ell$). Each of these chains will be common to both trees. Recall that, by Lemma 2.3, we may assume that every chain either survives or is atomized. The idea is that y-type chains and z-type chains are so long that they will all survive. The x-type chains are shorter and might be atomized. In fact, the x-type chains that are atomized will correspond to an FVS of D'.

The x-type and y-type chains will be used to encode the vertices of D', while the z-type chains will be used to make sure that no two of the x-type and y-type chains can be together in an agreement-forest component.

We build the trees $\mathcal{T}$ and $\mathcal{T}'$ as follows. For each vertex of D' of type v^- or v^+ we create an x-type chain. For each other vertex of D' we create a y-type chain. Finally, for each vertex and edge of the original graph D we create a z-type chain. All leaves of all chains have different labels. Now we combine the chains into two trees as follows.

First $\mathcal{T}$. Start with an arbitrary rooted binary tree on $|V|+|A|$ leaves labeled by the vertices and edges of D. We replace each leaf by its z-type chain, creating the tree which we call $\mathcal{T}_0$. Below each z-chain replacing a leaf labeled by an edge $(u,v) \in A$ we hang the y-type chain for u^v_{out}, and below that the y-type chain for v^u_{in}. Below each z-chain replacing a leaf labeled by a vertex $v \in V$ we hang an x-type chain for v^-, and below that an x-type chain for v^+.

Now $\mathcal{T}'$. Start with an arbitrary rooted binary tree on $2|V|$ leaves having two leaves for each vertex $v \in V$. Replace one of them by a concatenation of (from top to bottom) the y-type chains for $v^{u_1}_{\text{in}}, v^{u_2}_{\text{in}}, \ldots, v^{u_{d^-(v)}}_{\text{in}}$ and the x-type chain for v^-. Replace the other leaf for v by a concatenation of (from top to bottom) the x-type chain for v^+ and the y-type chains for $v^{w_1}_{\text{out}}, v^{w_2}_{\text{out}}, \ldots, v^{w_{d^+(v)}}_{\text{out}}$. Finally, hang a copy of $\mathcal{T}_0$ below the root. This concludes the construction of the MAAF instance. For an example, see Figures 5.2 and 5.3.

We claim that D' (and thus D) has an FVS of size at most f if and only if there exists an acyclic agreement forest of $\mathcal{T}$ and $\mathcal{T}'$ of size at most $1+2(|A|+|V|)+(\ell-1)f$.

To show this, consider the agreement forest A_D for $\mathcal{T}$ and $\mathcal{T}'$ in which $\mathcal{T}_0$ is one component, each x-type chain is one component, and each y-type chain is one component. Represent the components of A_D by their corresponding vertices and edges. The inheritance graph G_{A_D} is obtained from D' by adding the edges $(v^{u_i}_{\text{in}}, v^{u_j}_{\text{in}})$ for all $1 \le i < j \le d^-(v)$, the edges $(v^{w_i}_{\text{out}}, v^{w_j}_{\text{out}})$ for all $1 \le i < j \le d^+(v)$, and creating a vertex $v_{\mathcal{T}_0}$ corresponding to $\mathcal{T}_0$ with only outgoing edges to all other vertices. Hence $v_{\mathcal{T}_0}$ is not in any cycle of G_{A_D}. It is easy to see that any FVS of D', which by assumption has only vertices of type v^-, is also an FVS of G_{A_D}, and vice versa, since any directed cycle of G_{A_D} passing through an edge $(v^{u_i}_{\text{in}}, v^{u_{i+1}}_{\text{in}})$ or $(v^{w_i}_{\text{out}}, v^{w_{i+1}}_{\text{out}})$ must contain v^-. In addition, since v^--type vertices correspond to x-type chains, it is possible to make G_{A_D} acyclic by atomizing only x-type chains.

Let F be an FVS of D and let F' be the corresponding FVS of D'. Then we can construct an agreement forest $\mathcal{R}$ of $\mathcal{T}$ and $\mathcal{T}'$ as follows. One component consists of the tree $\mathcal{T}_0$. Each of the y-type chains is also one component, as well as each x-type chain that does not correspond to a vertex in F'. Finally, each x-type chain corresponding to a vertex in F' is atomized. Thus, the number of components is

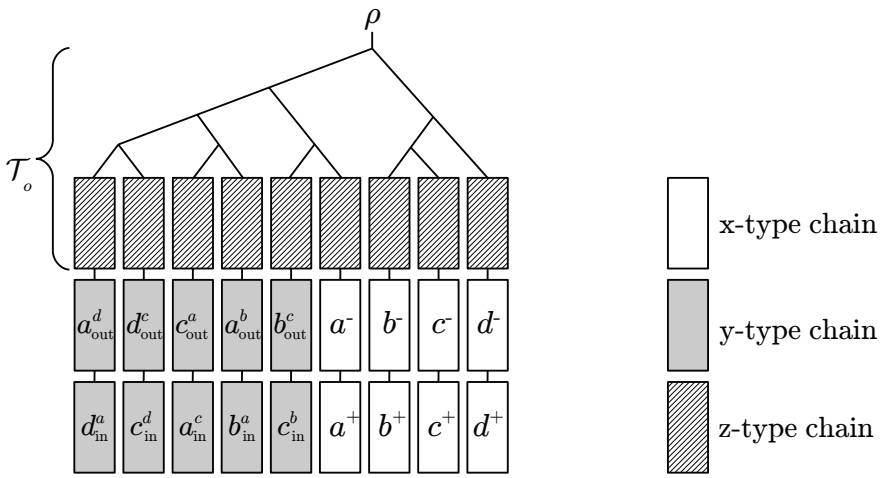

FIG. 5.2. *$\mathcal{T}$: the first tree of the constructed MAAF instance.*

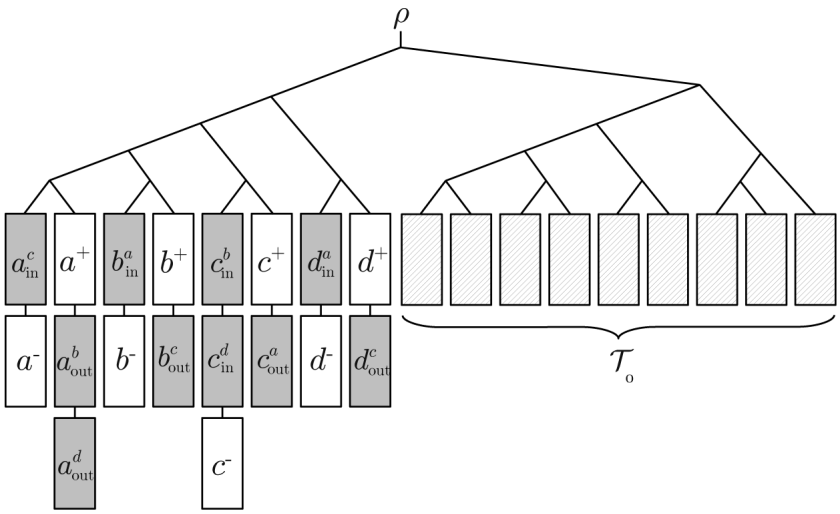

FIG. 5.3. *$\mathcal{T}'$: the second tree of the constructed MAAF instance.*

$1 + 2|A| + (2|V| - |F'|) + \ell|F'| = 1 + 2(|A| + |V|) + (\ell - 1)|F|$. We have to show that the inheritance graph $G_{\mathcal{R}}$ is acyclic. We can construct $G_{\mathcal{R}}$ from G_{A_D} as follows. Delete every vertex $v^- \in F'$ and instead add a vertex for each leaf of the corresponding x-type chain with incoming edges from $\mathcal{T}_0$ and from $v_{\text{in}}^{u_1}, v_{\text{in}}^{u_2}, \ldots, v_{\text{in}}^{u_{d^-(v)}}$. Since we only introduced leaves with incoming edges, this modification does not create any directed cycles. Thus, since F' contains a vertex of each directed cycle of G_{A_D}, and all vertices from F' have been removed, $G_{\mathcal{R}}$ is acyclic. It follows that $\mathcal{R}$ is an acyclic agreement forest for $\mathcal{T}$ and $\mathcal{T}'$.

To show the other direction, let $\mathcal{A}$ be an acyclic agreement forest of $\mathcal{T}$ and $\mathcal{T}'$. First, we may assume that all y-type chains and z-type chains survive in $\mathcal{A}$, since we will choose L sufficiently large (as will be specified later). To see this, recall that we may assume by Lemma 2.3 that each chain either survives or is atomized. Hence, if a

y-type chain or z-type chain does not survive, it is atomized and adds L components to the agreement forest. Second, observe that we may assume that all z-type chains are together in a single component (if they are not, we can put them together and reduce the number of components).

Now we argue that any pair of chains, at least one of which is not a z-type chain, cannot be together in a single component of $\mathcal{A}$. First, observe that if the root of one of the chains is an ancestor of the root of the other chain in $\mathcal{T}$, then the roots of the two chains are incomparable in $\mathcal{T}'$. Second, if the roots of the chains are incomparable in $\mathcal{T}$, then they are separated by a z-type chain in $\mathcal{T}$ but not in $\mathcal{T}'$. Hence, by (2) in the definition of an agreement forest, the two chains cannot be together in a single component of $\mathcal{A}$. Thus, the components of $\mathcal{A}$ are as follows. Tree $\mathcal{T}_0$ is the component containing the root and all z-type chains. Furthermore, each y-type chain, each surviving x-type chain, and each leaf of a nonsurviving x-type chain is a separate component. Let $\tilde{F}$ be the set of vertices of G_{A_D} corresponding to the nonsurviving x-type chains. Thus, each vertex in $\tilde{F}$ is of type v^- or v^+. We will show that $\tilde{F}$ is an FVS of G_{A_D} and hence of D'. We can construct $G_{\mathcal{A}}$ from G_{A_D} as follows. Remove each vertex in $\tilde{F}$ from G_{A_D} and add each leaf of the corresponding x-type chain as a separate vertex. Then add edges to these newly added vertices (these edges are not important since they do not create any directed cycles). Since $\mathcal{A}$ is an acyclic agreement forest, $G_{\mathcal{A}}$ is acyclic and hence $\tilde{F}$ is an FVS. The size $|\tilde{F}|$ of the FVS is equal to the number of nonsurviving x-type chains. Thus, $|\mathcal{A}| = 1 + 2|A| + (2|V| - |\tilde{F}|) + \ell|\tilde{F}| = 1 + 2(|A| + |V|) + (\ell - 1)|\tilde{F}|$.

The reduction is clearly polynomial time. It remains to show that it is approximation preserving. Suppose that there exists a c-approximation algorithm for MAAF. Say that m is the size of the MAAF returned by this algorithm and m^* the size of an optimal solution. Recall that MAAF minimizes the size of an acyclic agreement forest minus 1, so $m - 1 \le c \cdot (m^* - 1)$. We have shown that D has an FVS of size at most f if and only if $\mathcal{T}$ and $\mathcal{T}'$ have an acylic agreement forest of size at most $1 + 2(|A| + |V|) + (\ell - 1)f$. Thus, $m^* = 1 + 2(|A| + |V|) + (\ell - 1)f^*$. Moreover, an approximate solution f of DFVS can be computed from an approximate solution m of MAAF by taking $f = (m - 1 - 2(|A| + |V|))/(\ell - 1)$. Then we have

$$\begin{aligned} f &= \frac{m - 1 - 2(|A| + |V|)}{\ell - 1} \\ &\le \frac{c \cdot (m^* - 1) - 2(|A| + |V|)}{\ell - 1} \\ &= \frac{c(2(|A| + |V|) + (\ell - 1)f^*) - 2(|A| + |V|)}{\ell - 1} \\ &= c \cdot f^* + \frac{2(c - 1)(|A| + |V|)}{\ell - 1} \\ &\le c \cdot f^* + 1 \end{aligned}$$

if we take $\ell = \lceil 2(c-1)(|A| + |V|) + 1 \rceil$. We still need to specify the value of L, which needs to be sufficiently large so that all y-type chains and z-type chains survive. Since any graph trivially has an FVS of size $|V|$, any constructed MAAF instance has $m^* \le 1 + 2(|A| + |V|) + (\ell - 1)|V|$. Thus, a c-approximation algorithm will return an acyclic agreement forest of size m with $m - 1 \le c(m^* - 1) \le c(2(|A| + |V|) + (\ell - 1)|V|)$, and hence with $m \le c(2(|A| + |V|) + (\ell - 1)|V|) + 1$. So it suffices to take $L = \lceil c(2(|A| + |V|) + (\ell - 1)|V|) + 2 \rceil$.

Now take $\epsilon > 0$. If $f^* < 1/\epsilon$, we can compute an optimal solution for DFVS by

brute force in polynomial time. Otherwise, $1 \leq \epsilon \cdot f^*$ and we have

$$f \leq c \cdot f^* + \epsilon \cdot f^* = (c + \epsilon) f^*.$$

Thus, if there exists a c-approximation for MAAF, then there exists a $(c + \epsilon)$-approximation for DFVS for every fixed $\epsilon > 0$. ∎

In contrast to the result in section 4, the reduction above can only be used for constant c. It does *not* show that, e.g., an $O(\log |X|)$-approximation for MINIMUMHYBRIDIZATION would imply an $O(\log |V|)$-approximation for DFVS. Hence, it is indeed possible that MINIMUMHYBRIDIZATION admits an $O(\log |X|)$-approximation while DFVS does not admit an $O(\log |V|)$-approximation. For neither of the problems is such an approximation known to exist.

Finally, we note that Theorem 5.1 also allows us to improve upon the best-known inapproximability result for MINIMUMHYBRIDIZATION.

COROLLARY 5.2. *There does not exist a polynomial-time c-approximation for* MINIMUMHYBRIDIZATION*, where $c < 10\sqrt{5} - 21 \approx 1.3606$, unless* P = NP*. If the Unique Games Conjecture holds, then there does not exist a polynomial-time c-approximation for* MINIMUMHYBRIDIZATION *where $c < 2$.*

Proof. In [26] a simple reduction is shown from the problem VERTEX COVER to the problem DFVS. Specifically, given an undirected graph G as input to VERTEX COVER we create a directed graph G' by transforming each edge $\{u, v\}$ in G into two directed edges $(u, v), (v, u)$ in G'. It is easy to show that G' has an FVS of size k if and only if G has a vertex cover of size k. Consequently, any polynomial-time c-approximation algorithm for DFVS can be used to construct a polynomial-time c-approximation for VERTEX COVER. The latter problem does not permit a polynomial-time c-approximation, for any $c < 10\sqrt{5} - 21 \approx 1.3606$, unless P = NP [12, 13]. Also, it has been shown that if the Unique Games Conjecture is true, then no approximation better than 2 is possible [29]. Now, the proof of Theorem 5.1 shows that if there exists a c-approximation for MINIMUMHYBRIDIZATION, then there exists a $(c + \epsilon)$-approximation for DFVS for *every* fixed $\epsilon > 0$. Hence the existence of a c-approximation for MINIMUMHYBRIDIZATION where $c < 10\sqrt{5} - 21$ (respectively, $c < 2$) would mean the existence of a c'-approximation for DFVS (and thus also for VERTEX COVER) where $c' < 10\sqrt{5} - 21$ (respectively, $c' < 2$). ∎

REFERENCES

[1] V. BAFNA, P. BERMAN, AND T. FUJITO, *A 2-approximation algorithm for the undirected feedback vertex set problem*, SIAM J. Discrete Math., 12 (1999), pp. 289–297.

[2] M. BARONI, S. GRÜNEWALD, V. MOULTON, AND C. SEMPLE, *Bounding the number of hybridisation events for a consistent evolutionary history*, J. Math. Biol., 51 (2005), pp. 171–182.

[3] M. BARONI, C. SEMPLE, AND M. STEEL, *A framework for representing reticulate evolution*, Ann. Comb., 8 (2004), pp. 391–408.

[4] M. BONET AND K. S. JOHN, *On the complexity of uSPR distance*, IEEE/ACM Trans. Comput. Biol. Bioinformatics, 7 (2010), pp. 572–576.

[5] M. BORDEWICH, S. LINZ, K. S. JOHN, AND C. SEMPLE, *A reduction algorithm for computing the hybridization number of two trees*, Evolutionary Bioinformatics, 3 (2007), pp. 86–98.

[6] M. BORDEWICH, C. MCCARTIN, AND C. SEMPLE, *A 3-approximation algorithm for the subtree distance between phylogenies*, J. Discrete Algorithms, 6 (2008), pp. 458–471.

[7] M. BORDEWICH AND C. SEMPLE, *Computing the hybridization number of two phylogenetic trees is fixed-parameter tractable*, IEEE/ACM Trans. Comput. Biol. Bioinformatics, 4 (2007), pp. 458–466.

[8] M. BORDEWICH AND C. SEMPLE, *Computing the minimum number of hybridization events for a consistent evolutionary history*, Discrete Appl. Math., 155 (2007), pp. 914–928.

[9] J. Chen, Y. Liu, S. Lu, B. O'Sullivan, and I. Razgon, *A fixed-parameter algorithm for the directed feedback vertex set problem*, J. ACM, 55 (2008), 21.
[10] Z.-Z. Chen and L. Wang, *Hybridnet: a tool for constructing hybridization networks*, Bioinformatics, 26 (2010), pp. 2912–2913.
[11] J. Collins, S. Linz, and C. Semple, *Quantifying hybridization in realistic time*, J. Comput. Biol., 18 (2011), pp. 1305–1318.
[12] I. Dinur and S. Safra, *The importance of being biased*, in Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC '02), ACM, New York, 2002, pp. 33–42.
[13] I. Dinur and S. Safra, *On the hardness of approximating minimum vertex cover*, Ann. of Math. (2), 162 (2004), pp. 439–485.
[14] R. G. Downey and M. R. Fellows, *Parameterized Complexity*, Springer-Verlag, New York, 1999.
[15] G. Even, J. Naor, B. Schieber, and M. Sudan, *Approximating minimum feedback sets and multicuts in directed graphs*, Algorithmica, 20 (1998), pp. 151–174.
[16] J. Flum and M. Grohe, *Parameterized Complexity Theory*, Springer-Verlag, Berlin, 2006.
[17] O. Gascuel, ed., *Mathematics of Evolution and Phylogeny*, Oxford University Press, Oxford, 2005.
[18] O. Gascuel and M. Steel, eds., *Reconstructing Evolution: New Mathematical and Computational Advances*, Oxford University Press, New York, 2007.
[19] J. Gramm, A. Nickelsen, and T. Tantau, *Fixed-parameter algorithms in phylogenetics*, The Computer J., 51 (2008), pp. 79–101.
[20] J. Hein, T. Jing, L. Wang, and K. Zhang, *On the complexity of comparing evolutionary trees*, Discrete Appl. Math., 71 (1996), pp. 153–169.
[21] P. J. Humphries and C. Semple, *Note on the hybridization number and subtree distance in phylogenetics*, Appl. Math. Lett., 22 (2009), pp. 611–615.
[22] D. H. Huson, R. Rupp, V. Berry, P. Gambette, and C. Paul, *Computing galled networks from real data*, Bioinformatics, 25 (2009), pp. i85–i93.
[23] D. H. Huson, R. Rupp, and C. Scornavacca, *Phylogenetic Networks: Concepts, Algorithms and Applications*, Cambridge University Press, Cambridge, UK, 2010.
[24] D. H. Huson and C. Scornavacca, *A survey of combinatorial methods for phylogenetic networks*, Genome Biology and Evolution, 3 (2011), pp. 23–35.
[25] V. Kann, *On the Approximability of NP-Complete Optimization Problems*, Ph.D. thesis, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, 1992.
[26] R. M. Karp, *Reducibility among combinatorial problems*, in Complexity of Computer Computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, NY, 1972), Plenum, New York, 1972, pp. 85–103.
[27] S. M. Kelk and C. Scornavacca, *Constructing minimal phylogenetic networks from softwired clusters is fixed parameter tractable*, preprint, 2011; http://arxiv.org/abs/1108.3653.
[28] S. M. Kelk, C. Scornavacca, and L. J. J. van Iersel, *On the elusiveness of clusters*, IEEE/ACM Trans. Comput. Biol. Bioinformatics, 9 (2012), pp. 517–534.
[29] S. Khot and O. Regev, *Vertex cover might be hard to approximate to within* $2-\epsilon$, J. Comput. System Sci., 74 (2008), pp. 335–349.
[30] L. Nakhleh, *Evolutionary phylogenetic networks: Models and issues*, in The Problem Solving Handbook for Computational Biology and Bioinformatics, Springer-Verlag, New York, 2009.
[31] R. Niedermeier, *Invitation to Fixed Parameter Algorithms (Oxford Lecture Series in Mathematics and Its Applications)*, Oxford University Press, New York, 2006.
[32] C. H. Papadimitriou and M. Yannakakis, *Optimization, approximation, and complexity classes*, J. Comput. System Sci., 43 (1991), pp. 425–440.
[33] E. Rodrigues, M. Sagot, and Y. Wakabayashi, *The maximum agreement forest problem: Approximation algorithms and computational experiments*, Theoret. Comput. Sci., 374 (2007), pp. 91–110.
[34] C. Scornavacca, S. Linz, and B. Albrecht, *A first step towards computing all hybridization networks for two rooted binary phylogenetic trees*, J. Comput. Biol., to appear.
[35] C. Semple, *Hybridization networks*, in Reconstructing Evolution: New Mathematical and Computational Advances, Oxford University Press, New York, 2007.
[36] C. Semple and M. Steel, *Phylogenetics*, Oxford University Press, Oxford, 2003.
[37] P. D. Seymour, *Packing directed circuits fractionally*, Combinatorica, 15 (1995), pp. 281–288.
[38] L. J. J. van Iersel and S. M. Kelk, *When two trees go to war*, J. Theoret. Biol., 269 (2011), pp. 245–255.
[39] C. Whidden, R. Beiko, and N. Zeh, *Fast FPT algorithms for computing rooted agreement*

forests: Theory and experiments, in Proceedings of the 9th International Symposium on Experimental Algorithms (SEA 2010), Lecture Notes in Comput. Sci. 6049, Springer-Verlag, Berlin, 2010, pp. 141–153.

[40] C. Whidden, R. G. Beiko, and N. Zeh, *Fixed-parameter and approximation algorithms for maximum agreement forests*, preprint, 2011, http://arxiv.org/abs/1108.2664.

[41] C. Whidden and N. Zeh, *A unifying view on approximation and FPT of agreement forests*, in Algorithms in Bioinformatics, S. Salzberg and T. Warnow, eds., Lecture Notes in Comput. Sci. 5724, Springer-Verlag, Berlin, 2009, pp. 390–402.